UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
ESCUELA INGENIERIA DE SISTEMAS Y TELEMATICA

1. Datos generales

Materia: TEORÍA DE LA COMPUTACIÓN II
Código: FAD0212
Paralelo:
Periodo : Marzo-2019 a Julio-2019
Profesor: ACOSTA URIGÜEN MARIA INES
Correo electrónico macosta@uazuay.edu.ec

Docencia	Práctico	Autónomo:		Total horas
		Sistemas de tutorías	Autónomo	
4				4

Prerrequisitos:

Código: FAD0185 Materia: LENGUAJES DE PROGRAMACIÓN II

2. Descripción y objetivos de la materia

El estudio de la teoría de autómatas y de los lenguajes formales se ubica en el campo científico de la Informática Teórica, un campo clásico y multidisciplinar dentro de los estudios universitarios de Informática. Es un campo clásico debido no solo a su antigüedad - anterior a la construcción de los primeros computadores - sino, sobre todo, a que sus contenidos principales no dependen de los rápidos avances tecnológicos que han hecho que otras ramas de la Informática deban adaptarse a los nuevos tiempos a un ritmo vertiginoso. Es multidisciplinar porque en sus cimientos encontramos campos tan aparentemente dispares como la lingüística, las matemáticas o la electrónica.

El curso busca que el estudiante aprenda a realizar operaciones básicas con cadenas y con lenguajes; a construir modelos de autómatas para diferentes tipos de problemas y a reconocer diferentes tipos de lenguajes; utilizar métodos de generación de gramáticas formales; conocer el esquema fundamental de un traductor; aplicar los aspectos léxicos, sintácticos y semánticos para diseñar lenguajes formales y traductores, fundamento del diseño de compiladores. A partir de la comprensión de la jerarquía de Chomsky y su utilidad en el diseño de lenguajes de programación y sus traductores, el estudiante podrá evaluar la universalidad y limitaciones de la Máquina de Turing, modelo conceptual de la computadora contemporánea.

El/la ingeniero/a de sistemas y telemática debe estar en capacidad de comprender modelos de hardware y software, para lo cual la teoría de autómatas es un medio conceptual y metodológico de gran utilidad. Puede usarse para diseñar y probar el comportamiento de circuitos digitales básicos; para simular el analizador léxico de un compilador, es decir, el componente que separa el texto de entrada en unidades lógicas, tal como identificadores, palabra clave y otros signos. Igualmente, es uno de los fundamentos del diseño de software para explorar cuerpos de texto largos, como colecciones de páginas web, o para determinar el número de apariciones de palabras, frases y otros patrones.

3. Contenidos

1.	**Autómatas finitos y lenguajes regulares**
1.1.	Introducción: autómatas y compiladores (4 horas)
1.2.	Autómatas finitos y lenguajes regulares (4 horas)
1.3.	Autómatas finitos determinísticos (4 horas)
1.4.	Autómatas finitos no determinísticos (4 horas)
1.5.	Autómatas finitos con transiciones ¿ (4 horas)
1.6.	Expresiones regulares (4 horas)
1.7.	Propiedades de los lenguajes regulares (4 horas)
2.	**Autómatas de pila y lenguajes libres del contexto**
2.1.	Autómatas de pila (4 horas)
2.2.	Lenguajes libres del contexto (4 horas)
2.3.	Gramáticas libres del contexto (4 horas)
2.4.	Formas normales (4 horas)
3.	**Máquinas de Turing y**
3.1.	Máquinas de Turing (6 horas)

3.2.	Lenguajes recursivamente enumerables (2 horas)
3.3.	Lenguajes recursivos (2 horas)
3.4.	Tesis de Church (2 horas)
3.5.	Clasificación de Chomsky (4 horas)
3.6.	Universalidad y limitaciones de las máquinas de Turing (4 horas)

4. Sistema de Evaluación

Resultado de aprendizaje de la carrera relacionados con la materia

Resultado de aprendizaje de la materia	Evidencias
an. Genera modelos matemáticos y físicos para analizar y solucionar situaciones reales e hipotéticas presentados en la ingeniería de sistemas y telemática.	
- Evalúa la aceptabilidad de un lenguaje dado en un autómata determinado	-Evaluación escrita -Proyectos -Resolución de ejercicios, casos y otros
ap. Desarrolla la lógica algoritmica en el análisis y resolución de problemas aplicando los fundamentos de la programación.	
- Programa piezas de software en los que se sintetiza los algoritmos de funcionamiento de autómatas y reconocimiento de lenguajes formales	-Evaluación escrita -Proyectos

Desglose de evaluación

Evidencia	Descripción	Contenidos sílabo a evaluar	Aporte	Calificación	Semana
Evaluación escrita	Prueba escrita	Autómatas finitos y lenguajes regulares	APORTE 1	7	Semana: 6 (15/04/19 al 18/04/19)
Resolución de ejercicios, casos y otros	Resolución de ejercicios	Autómatas finitos y lenguajes regulares	APORTE 1	3	Semana: 6 (15/04/19 al 18/04/19)
Evaluación escrita	Resolución de ejercicios	Autómatas de pila y lenguajes libres del contexto	APORTE 2	5	Semana: 11 (20/05/19 al 23/05/19)
Evaluación escrita	Prueba escrita	Autómatas de pila y lenguajes libres del contexto	APORTE 2	5	Semana: 11 (20/05/19 al 23/05/19)
Proyectos	Proyecto: programación de ejercicios	Autómatas de pila y lenguajes libres del contexto, Autómatas finitos y lenguajes regulares, Máquinas de Turing y	APORTE 3	8	Semana: 15 (17/06/19 al 22/06/19)
Evaluación escrita	Prueba	Máquinas de Turing y	APORTE 3	2	Semana: 15 (17/06/19 al 22/06/19)
Evaluación escrita	EXamen escrito	Autómatas de pila y lenguajes libres del contexto, Autómatas finitos y lenguajes regulares, Máquinas de Turing y	EXAMEN	20	Semana: 19-20 (14-07-2019 al 20-07-2019)
Evaluación escrita	Examen escrito	Autómatas de pila y lenguajes libres del contexto, Autómatas finitos y lenguajes regulares, Máquinas de Turing y	SUPLETORIO	20	Semana: 20 (al)

Metodología

Las clases serán tanto magistrales como prácticas. Se llevarán a cabo explicaciones de contenidos y resolución de ejercicios dentro y fuera del aula. Por cada capítulo los estudiantes resolverán ejercicios prácticos.

Criterios de Evaluación

Para cada actividad valorada se presentará a los estudiantes los criterios específicos de evaluación a través de una rúbrica.

En las pruebas teóricas y en el examen se evaluará el conocimiento teórico del estudiante según la adecuada argumentación y razonamiento; se valorará también la iniciativa del estudiante mediante la diversidad de estrategias del planteo y resolución de problemas en una forma lógica y ordenada.

5. Referencias

Bibliografía base

Libros

Autor	Editorial	Título	Año	ISBN

Autor	Editorial	Título	Año	ISBN
Chiswell, Ian	Springer	A Course in Formal Languages, Automata and Groups	2009	
Fernandez, Maribel	Springer-Verlag	Models of Computation. An Introduction to Computability Theory	2009	
Maruoka, Akira	Springer	Concise Guide to Computation Theory	2011	
Parkes, Alan P	Springer	A Concise Introduction to Languages and Machines	2008	
Jurado Málaga, Elena	Universidad de Extremadura	Teoría de Autómatas y Lenguajes Formales	2008	
Gopalakrishnan, Ganesh	Springer	Computation Engineering. Applied Automata Theory and Logic	2006	

Web

Software

Bibliografía de apoyo

Libros

Web

Software

Docente

Director/Junta

Fecha aprobación: 06/03/2019

Estado: Aprobado